

Proyecto final del 1º trimestre



Índice

¿En qué consiste el proyecto?	3
Estructura de la colección	3
Consultas	4

¿En qué consiste el proyecto?

En este proyecto he realizado una base de datos que almacena información y características técnicas de una serie de teléfonos móviles. En esta base de datos hay una colección (“moviles”) compuesta de 10 documentos.

Además de la colección también hay un archivo donde hay varias consultas distintas entre sí donde se hace uso de operadores para filtrar búsquedas avanzadas que encuentran un móvil u otro, dependiendo del enunciado de la consulta. En este mismo archivo de consultas hay también unos métodos que no han sido vistos aún en clase.

Estructura de la colección

En la colección se ha usado el método “db.moviles.insertMany” para introducir varios documentos en la colección.

La colección está compuesta por 10 documentos, y cada documento incluye:

- Campos cuyo valor es una cadena de caracteres
- Campos cuyo valor es un número, entero o decimal.
- Campos cuyo valor es un array, tanto de números, como de cadenas de caracteres y también de documentos.
- Un campo cuyo valor es una fecha, en esta colección cada documento tiene un campo llamado “fechadelanzamiento” que registra la fecha de lanzamiento del teléfono móvil.
- Campos cuyo valor es otro documento que a su vez tiene distintos tipos de campos. Este tipo de campos también recibe el nombre de documentos embebidos.
- Campos cuyo valor es de tipo booleano, es decir, “true” o “false”, y “null”.

Consultas

```
db.moviles.find(  
  {  
    $and:[  
      {"precio.precio": {$lt: 150}},  
      {"marca": {$eq: "Xiaomi"}}  
    ]  
  }  
)
```

Esta consulta pretende buscar un teléfono móvil que cumpla con las dos condiciones:

- Que el precio sea inferior a 150€.
- Que la marca del teléfono sea Xiaomi.

El operador “\$and” obliga a que se cumplan las condiciones anteriormente descritas.

El operador “\$lt” obliga a que el valor del campo precio dentro del campo precio sea inferior a 150€ y el operador “\$eq” obliga a que el campo “marca” sea exclusivamente “Xiaomi”.

```
db.moviles.find(  
  {  
    $and:[  
      {fechalanzamiento: {$gte: new Date("2019-07-01"), $lte: new Date("2020-10-31")}},  
      {"hardware.bateria.capacidad":{$gte:4000}},  
      {"conectividad.nfc":true}  
    ]  
  }  
)
```

Esta consulta pretende buscar los móviles que cumplan con las siguientes condiciones:

- Que la fecha de lanzamiento se encuentre entre julio de 2019 y octubre de 2020.
- Debe tener una batería cuya capacidad sea mayor o igual a 4000 megaamperios.
- Que cuente con la tecnología NFC.

El operador “\$and” obliga a que se cumplan las condiciones anteriores. El operador “\$gte” obliga a que la fecha del campo “fechalanzamiento” sea posterior o igual a julio de 2019 y el operador “\$lte” hace que sea menor o igual a octubre de 2020. En la siguiente condición el operador “\$gte” obliga a que la capacidad de la batería sea mayor que 4000. Por último el campo “nfc” del campo “conectividad” debe tener el valor “true”.

```
db.moviles.find(  
  {  
    $and:[  
      {pais: {$ne:"China"}},  
      {"diseño.color": {$in: ["azul", "rojo"]}}  
    ]  
  }  
)
```

Esta consulta busca los móviles que no sea de una marca china y que sea de color azul o rojo.

Dentro del operador “\$and” usamos el operador “\$ne” para establecer que el campo “pais” no sea “China” y después usamos el operador “\$in” para que busque los móviles que en el campo “color” del campo “diseño” se encuentre el valor “azul” o “rojo”.

```
db.moviles.find(  
  {$or:[  
    {"diseño.alto":{"$gt:160}},  
    {"pantalla.pulgadas":{"$gt:7}}  
  ]}  
)
```

Esta consulta busca los móviles que tengan una altura de más de 160 milímetros o las pulgadas sean superiores a 7. El operador “\$or” obliga a que se cumplan alguna de las dos condiciones, es decir, encuentra los móviles que tienen una altura superior a 160 milímetros o los que tienen más de 7 pulgadas.

```
db.moviles.find(  
  {$and: [  
    {"diseño.materiales":{"$nin: ["plastico", "policarbonato"]}},  
    {"conectividad.banda":{"$in: ["5G", "4G"]}}  
  ]}  
)
```

Esta consulta busca los móviles que no sean ni de plástico ni de policarbonato y que tenga una banda 5G o 4G. El operador “\$and” obliga a que se cumplan las dos condiciones, después el operador “\$nin” hace que el campo “materiales” del campo “diseño” no sea ni “plastico” ni “policarbonato”. El operador “\$in” hace que el campo “banda” del campo “conectividad” valga o “5G” o “4G”.

```
db.moviles.find(  
  {$and: [  
    {"hardware.camara.dual": {$exists:true}},  
    {"hardware.camara.caracteristicasdual.resolucion":{"$gt:10}},  
    {"hardware.camara.caracteristicasdual.apertura":{"$gt:2}}  
  ]  
}  
)
```

Esta consulta busca los móviles que tengan cámara dual, que la resolución de esta sea mayor de 10 megapíxeles y que la apertura sea superior a 2. El operador “\$exists” hace que se busquen los documentos en los que existe ese campo. En este caso busca los documentos donde exista el campo “dual” dentro del campo “camara” dentro del campo “hardware”, además de las otras dos condiciones.

```

db.moviles.find(
  {$and: [
    {"conectividad.tipousb":{"$eq:"C"}},
    {$nor:[
      {"diseño.alto":{"$gt:155}},
      {"fechalanzamiento":{"$lte:new Date("2020-01-01")}}
    ]}
  ]}
)

```

Esta consulta busca los móviles que cumplan dos condiciones:

- Que el USB sea tipo “C”, usando el operador “\$eq”.
- Que la altura del móvil no sea superior a 155 o que la fecha de lanzamiento no sea inferior o igual al 1 de enero de 2020. Esto último lo permite el operador “\$nor” que hace que no se cumplan ni un valor ni el otro.

```

db.moviles.find(
  {$and: [
    {"hardware.almacenamiento.memoriainterna":{"$all: [64,128]"}},
    {"precio.precio":{"$lt:200}}
  ]}
)

```

Esta consulta busca los móviles que tengan exclusivamente las opciones de 64 y 128 gigabytes mediante el operador “\$all”, que busca los documentos en los que un campo tenga exclusivamente esos valores. También se debe cumplir que el precio sea inferior a 200€.

```

db.moviles.find(
  {$and: [
    {"hardware.almacenamiento.memoriainterna":{"$elemMatch: {$gte:100, $lte:550}}},
    {"pais":{"$regex: /estados unidos/i}}
  ]}
).pretty({$comment:"Esta consulta muestra los datos de una forma mas facil de ver en
Power Shell cuando usamos mongo. Si usamos mongo shell no es necesario"})

```

Esta consulta busca los móviles que cumplan dos condiciones:

- Que el campo “memoriainterna” del campo “almacenamiento” del campo “hardware” tenga un valor que se encuentre entre 100 y 550. El operador “\$elemMatch” filtra los valores que se encuentren en un rango especificado, en este caso entre 100 y 550.
- Que el valor del campo “pais” sea Estados Unidos, pero cuando no sabemos si ese valor lleva mayúsculas se utiliza el operador “\$regex /estados unidos/i”, la “i” es una opción para hacer insensibles las mayúsculas.

El comando \$comment permite hacer un comentario sobre algo, en este caso hago un comentario sobre la utilidad del método “.pretty()”.

```
db.moviles.find(
  {$and:[
    {pais:{$regex: /china/i}},
    {"diseño.huelladactilar":{$exists:true}}
  ]
})
).count({$comment:"Esta consulta cuenta el numero de moviles que son de una marca china y tienen huella dactilar"})
```

Esta consulta sirve para buscar cuántos móviles de una marca china tienen huella dactilar. El método “.count()” hace que nos muestre la información en un formato numérico, es decir, nos dice el número de documentos que cumplen con esa consulta.

```
db.moviles.find(
  {$and:[
    {"hardware.procesador.modelo":{$regex: /^Qualcomm/m}},
    {"hardware.bateria.capacidad":{$gte:3500}}
  ]}
)
```

Esta consulta busca los móviles que cumplan dos condiciones:

- Que el modelo del procesador sea de la marca Qualcomm. Para realizar esta búsqueda se utiliza el operador “\$regex: /^Qualcomm/m”. La “m” es una opción que nos permite utilizar el caracter “^” para que el valor de ese campo empiece por la palabra que hay entre las barras, y el caracter “\$” para que el valor de ese campo termine por la palabra que hay entre las barras.
- Que la capacidad de la batería sea mayor o igual que 3500.

```
db.moviles.find(
  {$and:[
    {fechalanzamiento: {$gte: new Date(2019-01-01)}},
    {"diseño.color": {$size:2}}
  ]}
)
```

Esta consulta busca los móviles que cumplan dos condiciones:

- Que la fecha de lanzamiento sea superior o igual a 2019.
- Que el móvil solo tenga dos opciones de color. Esto se hace utilizando el operador “\$size”, que hace que el campo “color” del campo “diseño”, que es un array de números, tenga exclusivamente 2 valores.

Aportación personal

A continuación voy a mostrar varios métodos y operadores que he investigado por mi cuenta.

```
db.moviles.distinct("modelo")
```

Este método hace que solo se muestren los modelos de cada móvil, es decir, el método “distinct” hace que solo aparezca el campo “modelo” de cada documento.

Resultado:

```
[
  { _id: 'Samsung Galaxy A30s', total: [ 4 ] },
  { _id: 'Huawei P50', total: [ 8 ] },
  { _id: 'Redmi Note 8', total: [ [ 3, 4 ] ] },
  { _id: 'Samsung Galaxy A22 5G', total: [ [ 4, 6, 8 ] ] },
  { _id: 'Redmi Note 10S', total: [ [ 6, 8 ] ] }
]
```

```
db.moviles.aggregate(
```

```
  [
    { $group: { _id: "$modelo", total: { $push: "$hardware.almacenamiento.memoriaram" } } },
    { $sort: { total: 1 } }
  ]
)
```

Esto hace que solo se vea el modelo y la memoria ram que lleva o puede llevar ese modelo de móvil. El método aggregate permite pasar la colección por una secuencia de filtros y nos devuelve la colección una vez pasados los filtros. El operador “\$group” agrupa y muestra documentos especificados por un “_id” y por cada distinto agrupamiento. Cada “_id” tiene un grupo único por valor. El “\$push” devuelve un array de los valores expresados de ese grupo.

Mediante el operador “\$sort” podemos ordenar el array que nos devuelve el método, en este caso está ordenado en sentido ascendente.

Resultado:

```
[
  { _id: 'Apple iPhone 7', total: [ 2 ] },
  { _id: 'Motorola Moto E20', total: [ 2 ] },
  { _id: 'Samsung Galaxy A30s', total: [ 4 ] },
  { _id: 'Alcatel 3 (2019)', total: [ 4 ] },
  { _id: 'Apple iPhone 13 Pro Max', total: [ 6 ] },
  { _id: 'Google Pixel 4a', total: [ 6 ] },
  { _id: 'Huawei P50', total: [ 8 ] },
  { _id: 'Redmi Note 8', total: [ [ 3, 4 ] ] },
  { _id: 'Samsung Galaxy A22 5G', total: [ [ 4, 6, 8 ] ] },
  { _id: 'Redmi Note 10S', total: [ [ 6, 8 ] ] }
]
```



```

db.moviles.aggregate(
[
  {$match: {marca: {$in: ["Xiaomi", "Huawei", "Samsung"]}}},
  {$group: {_id: "$modelo", total: {$push: "$hardware.almacenamiento.memoriaram"}}},
  {$sort: {total:1}}
]
)

```

Esto hace lo mismo que el anterior ejemplo, pero esta vez solo de las marcas Xiaomi, Huawei y Samsung. Esto se hace añadiendo el operador \$match, que hace que el grupo agregado coincida con alguna de esas marcas, añadiendo el operador \$in y las marcas anteriores.

Resultado:

```

[
  { _id: 'Samsung Galaxy A30s', total: [ 4 ] },
  { _id: 'Huawei P50', total: [ 8 ] },
  { _id: 'Redmi Note 8', total: [ [ 3, 4 ] ] },
  { _id: 'Samsung Galaxy A22 5G', total: [ [ 4, 6, 8 ] ] },
  { _id: 'Redmi Note 10S', total: [ [ 6, 8 ] ] }
]

```